REMARKS

This amendment is responsive to the Office Action dated August 30, 2006. Applicant has amended claims 1-3, 8, 14, 15, 17, and 24-26. Claims 1-5, 7-8, 11-26 are pending.

Claim Objections

In the Office Action, the Examiner objected to claims 24 and 25 because of the following informalities: claim 24 should contain --:-- after "the method comprising" and claim 25 should contain --:-- after "HTTP header by." Applicant has amended claims 24 and 25 to correct the informalities.

Claim Rejection Under 35 U.S.C. § 101

In the Office Action, the Examiner rejected claims 1-26 [sic] under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Specifically, the Examiner stated that claims 1-26 [sic] fail to produce a tangible result because the step of comparing alone does not produce such a result.

As a preliminary comment, Applicant presumes the Examiner meant to refer to pending claims 1-5, 7-8, 11-26 and not claims 1-26.

Applicant has amended independent claim 1 to recite comparing the predefined flag and a result of the bitwise operation to produce an indication for a case-insensitive string match, wherein the indication for the case-insensitive match indicates whether all characters of the unknown string within the network message match all corresponding characters of the identified predefined string so as to match one of the known headers of the network communication protocol. Also, amended claim 1 now specifically recites processing the network message based on the indication of the case-insensitive match, and outputting a response from the network device based on the processed network message.

In this manner, the result of the comparison is used to control processing of the network message and produce the tangible result of a response to that message. Support for the amendment can be found on pg. 6, ln. 19 – pg. 7, ll. 6, which states that string matching module 24 reduces time to transfer data between clients 12 and servers 24, and that the servers process headers of an HTTP message so that the server can appropriately respond to the request. Applicant submits that the pending claims are now directed to statutory subject matter in that the

claims specifically recite a concrete, tangible and useful result of processing network messages and producing responses. Applicant respectfully requests withdrawal of the rejection.

Claim Rejection Under 35 U.S.C. § 112

In the Office Action, the Examiner rejected claims 1, 24, and 25 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Applicant has amended claims 1, 24 and 25 for purposes of clarification. Applicant submits that claims, as amended, particularly point out and distinctly claim the subject matter, as required by 35 U.S.C. 112, second paragraph.

Claim Rejection Under 35 U.S.C. § 103

In the Office Action, the Examiner rejected claims 1-5, 7, 8, 11-20, 22 and 23 under 35 U.S.C. 103(a) as being unpatentable over a combination of Branstad (USPN 6,842,860) in view of HTTP 1.1, Fielding et al., June 28, 2001, pages 1-6, Chapter 3 Protocol Parameters", pages 1-10, Chapter 4 HTTP Message, pages 1-4 ("Fielding"), Smith et al. (US 6,377,991) and "Official Notice" in view of Slater et al. (US 6,654,796) and James et al. (US 6,523,108)

The applied references fail to disclose or suggest the inventions defined by Applicant's claims, and provide no teaching that would have suggested the desirability of modification to arrive at the claimed invention.

As one example, Applicant's amended claim 1 recites storing, on a network device, a database containing a plurality of predefined strings, wherein the predefined strings stored within the database represent known headers for a network communication protocol. Amended claim 1 further requires receiving, with the network device, a network message, and in response to receiving the network message, selecting one of the plurality of predefined strings stored within database of the network device.

Prior to this amendment, the Examiner asserted that Branstad teaches a predefined string at col. 3, ll. 26-38. As set forth in Applicant's previous communications, Branstad at col. 3, ll. 26-38 states that a network device "generates" an authentication tag 140 from a packet that is being sent. Applicant's current amendments distinguish Branstad and make clear that claim 1 requires storing, on a network device, a plurality of pre-defined strings within a database. Thus,

the pre-defined string recited in Applicant's claim 1 must be selected from a database that stores a plurality of pre-defined strings and, therefore, cannot be a portion of the network message received by the device, as suggested by the Examiner on page 3 of the Office Action. Rather, Applicant's claim 1 makes clear that one of the pre-defined strings is selected from the database in response to that device receiving a network message.

To be clear, Branstad's authentication tag is generated based on the contents of a network packet and does not represent a pre-defined string selected from a database in response to a receive message, i.e., after a message is received, as required by claim 1. Branstad provides no teaching of such features. Rather, the Branstad authentication tag is dynamically computed based on the particular contents of the packet being sent.

Further, Applicant's claim 1 requires that the predefined strings stored within the database represent "known headers for a network communication protocol." Branstad's authentication tag (which is a string compared to a different authentication tag in Branstad) is generated (i.e., computed) from the contents of a network packet using cryptographic techniques.[1] In no manner can the cryptographic authentication tags generated by Branstad based on the content of packets be considered predefined strings that represent known headers for a network communication protocol, as required by claim 1. To the contrary, even in view of the other cited references, any string generated by a cryptographic process simply cannot be a known header for a communication protocol.

Applicant's claim 1 further requires identifying a portion of the network message as an unknown string for comparison with the selected predefined string. Thus, the literal language of claim 1 limits this claimed embodiment to a method where the predefined string is selected from a database of predefined strings in response to the received message, i.e., after a message is received, and that the unknown string is a portion of that received network message. These features are not taught or suggested by the combination of references cited by the Examiner.

Furthermore, Applicant's amended claim 1 requires performing a bitwise exclusive OR operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string. Therefore, claim 1 literally requires that a bitwise exclusive OR operation be performed between:

[1] Branstad at col. 21, ll. 4-46,

(1) a segment of the unknown string that was identified within a network message, and (2) a segment of the predefined string that was selected from a database of predefined strings in response to the message. It should be apparent to the Examiner how these elements now distinguish the claim from the prior art.

For example, the Examiner's rejection of claim 1 in its previous form is based on an interpretation that claim 1 reads on Branstad in that Branstad teaches application of an exclusive OR operation to the received message to compute the authentication tag. Based on the recent Office Action, it is clear that the Examiner bases this conclusion on an interpretation that the XOR operation in Branstad to different portions of the network message can be viewed as an application to two strings, i.e., the pre-defined string and the unknown string. Amended claim 1 now clearly distinguishes over this interpretation by explicitly stating that bitwise exclusive OR operation be performed between: (1) a segment of the predefined string that was selected from a database of stored predefined strings, and (2) a segment of the unknown string that was identified as a portion within a received network message.

In addition, Applicant's claim 1 is limited to performing a bitwise exclusive OR operation to the two strings for which a case-insensitive match is being generated, i.e., between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the predefined string. In other words, the XOR operation of claim 1 is applied to the strings being compared, and those strings being compared are the inputs to that XOR operation. Claim 1, as a whole, requires performing a bitwise operation on a predefined flag and a result of the XOR operation, and comparing the predetermined flag and a result of the bitwise OR operation to produce an indication for the case-insensitive string match. The combination of references cited by the Examiner fails to teach or suggest this process for numerous reasons.

First, none of the references, either singularly or in combination, describe any method that determines whether an unknown string matches a predefined string by performing an XOR operation between the predefined string and the unknown string. As stated in Applicant's previous response, the Examiner's reasoning is logically flawed in asserting that Branstad teaches performing a bitwise exclusive OR operation between at least a segment of the predefined string and a segment of the unknown string when comparing those two strings.

Branstad teaches application of an XOR operation to portions of a message so as to **compute** a binary string, i.e., cryptographic authentication tag. Branstad makes abundantly clear that the XOR operation is not used as part of a comparison operation of the two strings that are provided as inputs to the XOR operation, as required by claim 1. Rather, Branstad describes application of XOR operations only for computing an authentication tag. Only after the tag is generated using cryptographic techniques (which may involve an XOR) is it then compared in a conventional way to a different string, i.e., the authentication tag extracted from the received message.

For example, in col. 3, ll. 35-43, Branstad makes clear that an authentication tag is computed from an inbound message (by use of cryptographic techniques as Branstad explains), and only after computing the authentication tag from the message is a comparison operation performed to compare that computed authentication tag to an a different authentication tag extracted from the same message:

> Upon receipt of communication **150**, receiver **120** extracts message **130'** and authentication tag **140'**. The extracted message **130'** is used by authentication tag computation module **122** in receiver **120** to produce authentication tag **140"**. A comparison is then made to determine if the generated authentication tag **140"** matches the extracted authentication tag **140'**. If the authentication tags match, then message **130'** is authenticated.

Thus, in no manner does Branstad teach or suggest performing a bitwise XOR operation between a predefined string and an unknown string so that the result of that XOR operation could be used as an indication of a match between those two strings that were applied as inputs to that XOR operation, as required by claim 1.

In other words, the Examiner's argument is based on the premise that different portions of the received message in Branstad could be viewed as two strings, and that Branstad suggests applying an XOR operation to those two strings when computing the cryptographic authentication tag. Even assuming this is a valid interpretation of Branstad, in no way does Branstad in view of the other cited references utilize an XOR operation in a process where the result of the XOR operation could be used to provide an indication that a case-insensitive match exists between those two input strings or the XOR operation, as required by claim 1.

The Examiner's combination of references would require that the XOR operation in Branstad produce a result that could somehow be used to indicate a case-insensitive match between the two portions of the received message to which the XOR operation is applied. This clearly is not the case. The XOR operation in Branstad is applied to the data portion of the message to cryptographically compute an authentication tag that can subsequently be compared to a different authentication tag. The XOR operation is not used to produce a result that somehow determines whether the portions of that message applied as inputs to the same XOR operation match. The Examiner' interpretation of Branstad as suggesting these elements of Applicant's previously pending claim 1 is incorrect. Nevertheless, Applicant's clarifying amendments submitted herein clearly distinguish over such an interpretation.

The combination of references cited by the Examiner fail to address this flaw in the Examiner's reasoning with regard to Branstad. Fielding merely describes the HTTP protocol and, in particular, the parameters and headers used when communicating via the HTTP protocol. Fielding does nothing to overcome the fundamental deficiencies of Branstad. Modification of the Branstad authentication technique in view of Fielding, as suggested by the Examiner, would result only in an authentication system in which the messages sent over the network conform to the HTTP protocol. The cryptographic technique used in Branstad to generate the authentication tag from an HTTP message would be the same. To the extent the cryptographic technique (e.g., KR5) uses XOR operations and rotations, the operations would be applied to the HTTP message to compute the authentication tag in the system proposed by the Examiner. Fielding provides no suggestion to use an XOR operation for any other purpose, and the combination of Branstad in view of Fielding fails to teach or suggest applying a bitwise XOR operation between two different messages when determining whether the strings match.

In rejecting claim 1, the Examiner also cites Smith as teaching the "well-known" concept of applying a predefined flag. However, Branstad in view of Fielding and Smith also fail to teach or suggest Applicant's claim 1 for several reasons.

First, at the cited portion, Smith describes "multiplying" the results on an XOR operation by a constant. Branstad in view of Smith does not teach or suggest performing a bitwise operation between a predefined flag and a result of the exclusive OR operation, and then

comparing the predetermined flag and a result of the bitwise OR operation to produce an indication for the case-insensitive string match, as required by claim 1.

Second, modification of the Branstad authentication technique in view of Smith makes little or no sense. The result of the XOR operation in Branstad is the generation of the cryptographic authentication tag. In rejecting claim 1 over Branstad in view of Fielding and Smith, the Examiner is proposing to modify the Branstad system so that a predefined flag is applied to the result of the XOR operation, which in Branstad is the authentication tag. The Examiner has failed to explain why one would multiply the authentication tag of Branstad with a constant, as taught by Smith, and how such an operation would in anyway be useful in comparing strings. In fact, the predefined flag (i.e., the "constant") in Smith is used in a hashing function to spread URLs across a hash space. The combination of Branstad in view of Fielding and Smith fails to provide any suggestion as to how a predefined flag could be used with a result of an XOR operation between strings in any manner that would aid detecting a match between two strings.

Third, The Examiner has simply offered no explanation as to how the resultant system could actually achieve a case-insensitive comparison between strings after incorporating such an operation within Branstad. The Applicant is at a loss as to how the modification proposed by the Examiner could even be achieved. The Examiner appears to gloss over the fact that Branstad describes techniques for checking the authentication of a sender by comparing cryptographically generated authentication tags. This cryptographic comparison for purposes of authentication, presumably, would not tolerate matching of characters of different cases, e.g., "a" and "A." It is highly unlikely that a binary digital signature, for example, would be accepted as a match even though it was different from the expected binary string. Moreover, no actual string comparison function is discussed in Branstad and the comparison function can only be concluded as a conventional string comparison technique.

The Examiner's rejection of claim 1 appears to be nothing more than piecemeal mapping of Applicant's claim language to an aggregation of prior art references combined in a nonfunctional manner. There is no evidence whatsoever that the techniques taught by the references could in any manner be employed to produce a case-insensitive match between two strings, as suggested by the Examiner.

Fourth, with respect to motivation for incorporation of these functions, the Examiner refers the Applicant to a printout of archived messages between Rob Soccacio and Eric Sit discussing a "case insensitive compare when getting environment headers." Even assuming this reference constitutes prior art, the "fix" proposed by the messages actually specifically includes software that invokes a conventional string compare function call (**strnicmp**) to compare each string. Thus, the messages cited by the Examiner actually suggest a conventional approach that is the antithesis of Applicant's technique. Applicant does not understand how the Examiner can rely on this reference for support for his Official Notice or as motivation to aggregate the references in the manner proposed by the Examiner.

For at least these reasons, modification of Branstad in view of Fielding, Smith and Official Notice fails to establish a prima facie case for non-patentability of Applicant's claims under 35 U.S.C. 103(a). Withdrawal of this rejection is requested.

CONCLUSION

All claims in this application are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of all pending claims. Please charge any additional fees or credit any overpayment to deposit account number 50-1778. The Examiner is invited to telephone the below-signed attorney to discuss this application.

Date: November 30, 2006

By: Kent Sieffert

Name: Kent J. Sieffert
Reg. No.: 41,312

SHUMAKER & SIEFFERT, P.A.
8425 Seasons Parkway, Suite 105
St. Paul, Minnesota 55125
Telephone: 651.735.1100
Facsimile: 651.735.1102